



**Rafael Alberti**  
Instituto de Educación Secundaria

Trabajo final de grado

DAW

MASTER'S TRICK

Bibliografía Muriel Poussa

# Índice

1. Índice.....	2
2. Introducción.....	3, 4
3. Descripción.....	4, 5
4. Instalación.....	5, 6
5. Guía de estilo y prototipado.....	6, 8
6. Diseño .....	9
7. Desarrollo.....	10, 11
8. Pruebas.....	12
9. Despliegue.....	12
10. Manual.....	12, 14
11. Conclusiones.....	14, 15
12. <u>Índice</u> de tablas.....	16
13. Bibliografía .....	17

## Introducción

La idea inicial del proyecto aparece a raíz de mi función como master en las partidas en mi círculo personal, por experiencia propia la peor parte de una partida de rol es el momento de tener que estar explicado a cada jugador como crear su personaje, estar pendiente a que cada jugador rellene correctamente su ficha.

En las primeras partidas puede resultar fácil de hacer, pero cuando ya llevas un tiempo jugando rol resulta una tarea muy tediosa, que con facilidad consume cerca de dos horas o más dependiendo del número de personas.

Las comunidades de algunos juegos han creado programas de pc para crear fichas de personajes como fue el caso de la leyenda de los cinco anillos, aplicación la cual ya no existe, y no hace muchos años el juego mas famoso dragones y mazmorras creo su propia herramienta online de creación de personajes [DNDbeyond](#). También remarcar que esta idea tomo forma por conciencia medioambiental ya que, aunque no lo parezca la cantidad de residuos de papel que se pueden generar puede resultar asombrosa.

Sin embargo, son muchos los juegos que no cuentan ni siquiera con un api donde consultar datos básicos del juego y poder retroalimentar una aplicación.

La idea inicial de este proyecto no es surtir soporte solamente a DND, sino que los usuarios puedan cerrarse personajes de una amplia gama de juegos como aquelarre, dark hersy, la llamada, etc. Permitiendo mantener un histórico de los personajes, ya sea que estén en partidas en curso, partidas finalizadas o muertos en campaña, el poder centralizar todos tus personajes en una única aplicación y que esta te permita actualizarlos, puede reducir en gran medida la generación de recursos, además que como master de esta forma las fichas no serían susceptibles de perderse o estropearse.

A muy largo plazo seria ideal poder brindar soporte directo para las partidas de rol, permitiendo que los jugadores utilicen salas con cuadrícula donde poder cargar los mapas de sus aventuras.

Como tecnologías me decane por utilizar java spring boot para el backend, angular para todo el desarrollo del frontend y utilizar Docker para mantener desplegada la base de datos.

#### Antecedentes:

Las webs que yo he llegado a utilizar y que siempre me a parecido que les faltaba algo son las siguientes:

Dndbeyond: aplicación web creada por los mismos creadores de DND, poco intuitiva y bastante densa de utilizar.

Dungeon master`s valt: aplicación web creada por aficionados a dnd, bastante simple de utilizar, pero poco actualizada.

Carácter builder: aplicación web bastante esquemática y lenta de utilizar.

Rold20: aplicación web mayoritaria, no brinda la capacidad para crear un personaje solo es de soporte para las partidas, pero da la opción de guardar el documento de tu ficha de personaje.

#### Descripción:

La aplicación resultante ha sido una herramienta web para crear y almacenar personajes del libro básico de dnd, dado que este juego tiene un sistema de creación de personajes muy muy simple, al ser un juego muy extendió y reconocido ha sido bastante fácil encontrar un api que brindara los suficientes datos y estos definidos de tal forma que son realmente utilizables.

Algunas Apis que pude encontrar como [dnd5eapi](#), son Apis excesivamente simples que no brindan datos funcionales. por ello utilizo el api [open5](#).

Las funcionalidades que he logrado han sido, la creación de cuneta de usuario, con su respectivo login y eliminación de cuneta, y la creación de personajes de dnd almacenados y asociados a un usuario.

## Instalación:

Node:

Para poder instalar angular y muchas de las dependencias necesarias, se tiene que hacer uso de npm por lo que tendremos que instalar node descargándolo desde su página web.

<https://nodejs.org/es/download/>

Angular:

Una vez instalado node podremos instalar angular mediante la línea de comando, ya sea la propia cmd o la propia consola de node, si los parámetros están bien definidos podremos empelar node desde la cmd e incluso desde el mismo visualstudio. Con el siguiente comando se instala angular.

```
npm install -g @angular/cli
```

una vez instalado podremos generar nuestro proyecto mediante:

```
ng new my-first-project
```

la creación del proyecto puede tardar mas que la instalación del mismo angular, una vez creado tendremos que movernos al proyecto y ejecutar

```
ng serve
```

De esta forma se levantará el servidor de angular que por defecto se ejecuta en el puerto 4200

Spring boot:

Las dependencias utilizadas en este proyecto han sido:

Jpa ,Spring boot web, H2, Lombok, Hibernate

Github:

La el desarrollo a estado todo el tiempo respaldado con repositorios en GitHub haciendo uso de la línea de comando de git.

<https://github.com/edMP/autoDND>

## Guía de estilos y prototipado:

### Login

La web inicia directamente en el login, botón login deshabilitado y botón para ir a registrarse

El prototipo muestra una interfaz de usuario con un encabezado gris que contiene el texto "titulo de la web". En el centro de la página hay un formulario gris con tres campos de entrada blancos: "usuario", "contraseña" y "login". Debajo del campo "login" hay un botón "registrarse".

# Registro de usuario

Se solicitará Nick de usuario, datos personales como nombre y apellidos, contraseña y email. El botón esta deshabilitado hasta que se completen todos los campos



nick

nombre de usuario

apellido

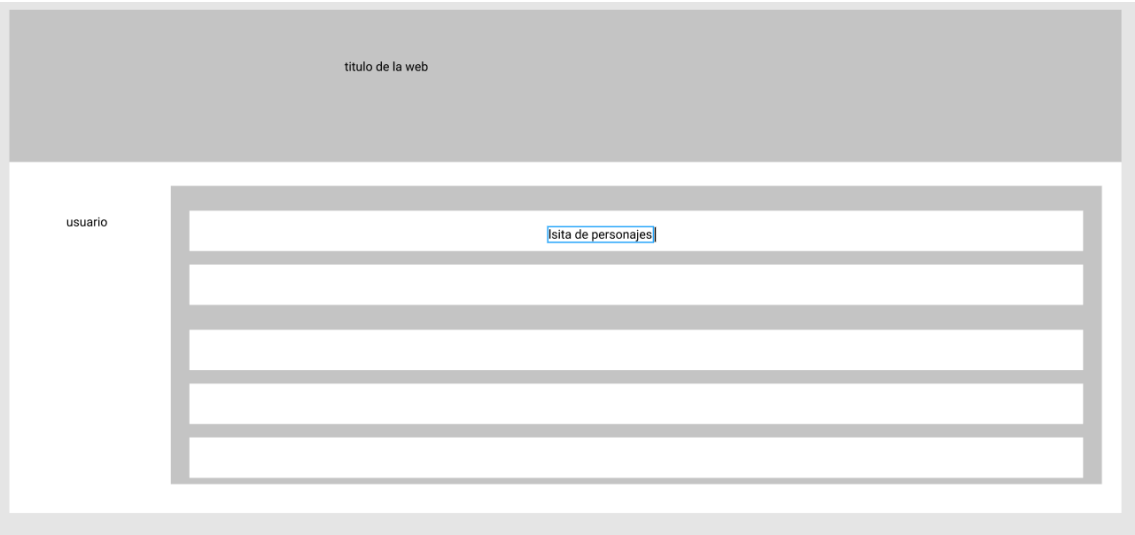
contraseña

email

registrarse

# Panel de usuario

El panel de usuario mostrar una lista con todos los personajes que tiene el usuario logueado, se mostrar nombre ,nivel y clase.



## Creación de personaje

La creación de personaje es lo mas automática posible, los equipamientos y hechizos aparecerá un desplegable para ir añadiéndolos al personaje, el trasfondo será tato seleccionable de algunos predefinidos o el usuario podrá insertar el suyo propio.

Diagrama de la interfaz de usuario para la creación de personaje. El encabezado contiene el "titulo web" (157 x 42). El cuerpo de la interfaz incluye:

- Un campo con los atributos del personaje auto generados con la vista previa de como quedarían los bonificadores de tiradas.
- Un botón para crear los atributos.
- Formas de entrada para: nivel y edad, nombre, raza, clase, sexo, alineamiento, trasfondo, equipamiento y hechizos.
- Una nota: "mediante pop ups se muestran opciones de equipo y hechizos".

## Panel de personaje

Se mostraran los datos del personaje, y solo será editables algunos como desventajas equipo o las estadísticas.

Diagrama de la interfaz de usuario para el panel de personaje. El encabezado contiene el "titulo de la web". El cuerpo de la interfaz incluye:

- Un panel "personaje" que muestra datos que no pueden cambiar: nombre, raza, clase, trasfondo, historia.
- Un panel "estadísticas".
- Un panel "equipo".
- Un panel "desventajas" (65 x 14).



## Diseño:

La aplicación tendrá dos entidades base, usuario y personaje.

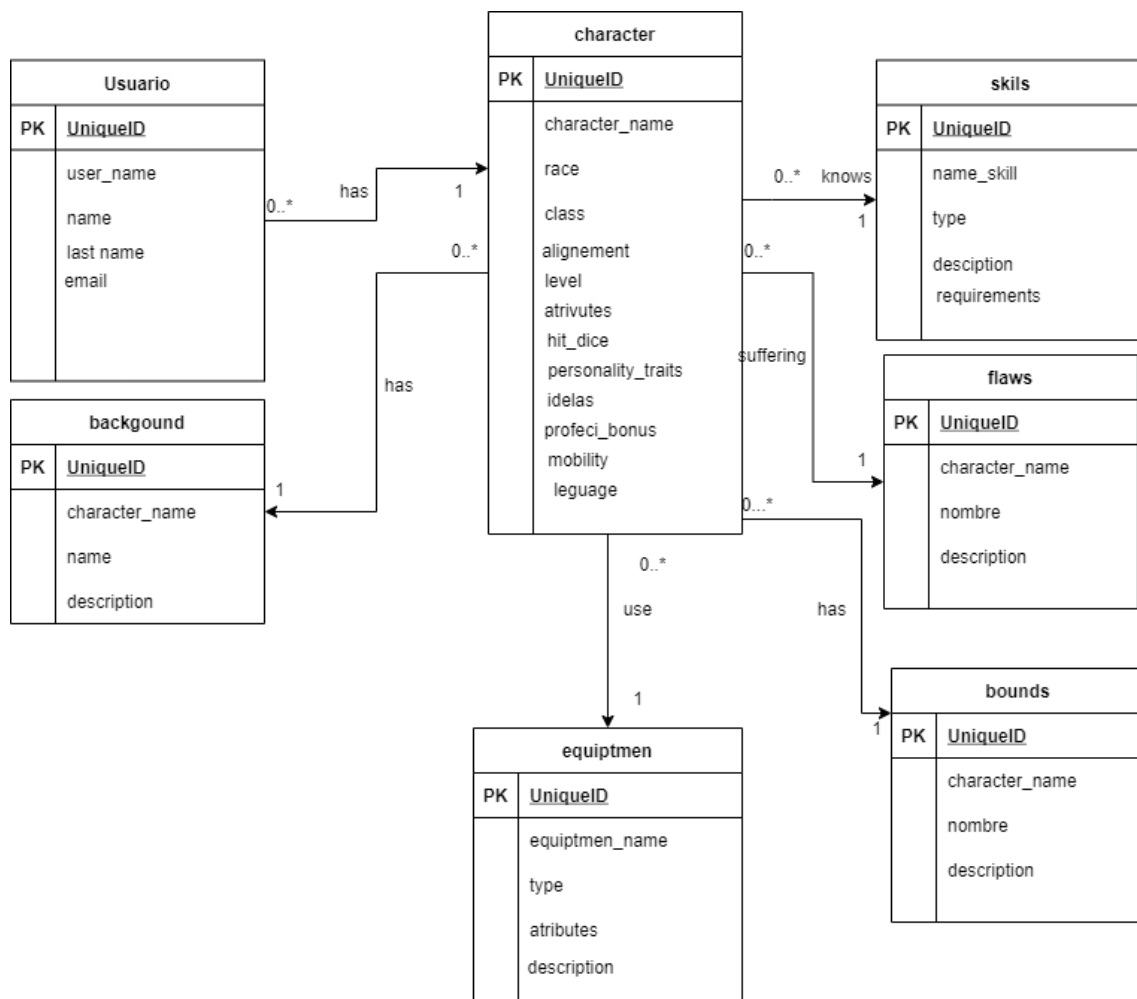
Usuario: datos del usuario

- Nick
- Datos personales
- Personajes

Personaje:

Esta tabla se encuentra relacionada con todos aquellos datos que pueden funcionar como una entidad propia, ej los equipos ya que estos tienen sus propias descripciones y estadísticas.

Diagrama general



## Desarrollo:

El desarrollo del proyecto ha resultado muchísimo mas complejo de lo que cabía esperar, la mayor complejidad que he podido tener ha sido en la organización del proyecto y como empezarlo, por mas diagramas o plantillas que define no veía la forma de arrancar con el proyecto.

Dentro de la organización cosas como, definir donde se realizarían las consultas a la api y que api utilizar han retrasado en gran medida el inicio del proyecto.

Respecto a las herramientas utilizadas angular es un framework bastante difícil de comenzar a trabajar la distribución de componentes y su comunicación es muy difícil de comprender y más aun de ganar soltura, si no llega a ser por su uso durante las practicas no habría sino ni capaz de definir una plantilla. No por ello me arrepiento de implementarlo a pesar de ser una herramienta muy densa de utilizar da una serie de elementos, que permite generar una herramienta bastante estilizada, practica y segura.

Para trabajar en angular he utilizado visual studio code como editor, una herramienta que no presenta ninguna dificultad para trabajar.

El desarrollo del backend lo he realizado en intellij, este ide aporta una gran cantidad de facilidades como tener integrado un spring boot initializer.

El desarrollo en si comenzó por el backend, definiendo todos los modelos y los controladores de cada uno, mediante el uso de insomnia se ha podido ir ejecutando cada controlador. Y precisamente el ir probando cada controlador y viendo como el formato y estructura en que se devuelven las consultas me a servido para poder predefinir y hacerme una idea de cómo crear el front.

Una vez logre hacer que la mayoría del back funcionara como yo pretendía, la creación de frontend ha sido algo más simple, a pesar de tener una idea aproximada de como querer que se viera mediante las maquetas la creación y diseño del frontend me ha parecido de los momentos más difíciles del desarrollo.

Carezco completamente de la capacidad necesaria para poder pre estructurar un front y poder definirlo en estructura, es decir la capacidad de pensar en una serie de métodos con funcionalidad y su estructura no soy capaz de extrapolarlo en la hora de definir una interfaz y su funcionalidad.

Sobre Angular he de remarcar, que me ha resultado una herramienta muy complicada de utilizar, por lo que tener que montar un proyecto mediano ha sido bastante difícil sobre todo por la estructura que tiene la herramienta en sí.

Gracias a que en las practicas recibí algún curso intensivo de Angular ha sido algo mas liviano, pero de igual manera identificar un fallo o no perderse en su estructura de carpetas es realmente muy difícil.

Ahora bien, Angular ofrece una gran variedad de implementaciones que facilitan en gran medida el trabajo, claro esta por ello muchas cosas es mas sencillo instalar la librería necesaria que buscar la forma de hacerlo de otra manera, esto puede hacer que la aplicación se muy pesada. Por otra parte uniendo Angular y Bootstrap el diseño de la interfaz se hace increíblemente más sencillo, además de poder hacer validaciones de cosas simples en un solo instante.

La decisión más importante que tomé y la única forma que vi de poder comenzar con el proyecto fue sacrificar la multifuncionalidad del proyecto, es decir enfocarme solo y exclusivamente en un juego de rol.

También se decidió sacrificar escalabilidad de la aplicación al nivel de los modelos, uniendo las tablas de atributos como elementos propios del personaje, con esta decisión he podido tener un desarrollo mas liviano y en gran medida facilitar el arranque del desarrollo ya que podía definir una estructura simple que me permitiese comenzar a codificar.

Como control de versiones se ha utilizado GitHub.

## Pruebas:

Gran parte de las pruebas se han realizado en Insomnia, documento que se adjuntara junto a este, ya que esta herramienta permite una vista rápida de las peticiones realizadas, a la hora de la funcionalidad web se ha intentado introducir manualmente la mayor cantidad de posibilidades de datos para poder brindar una buena cobertura de errores.

## Despliegue:

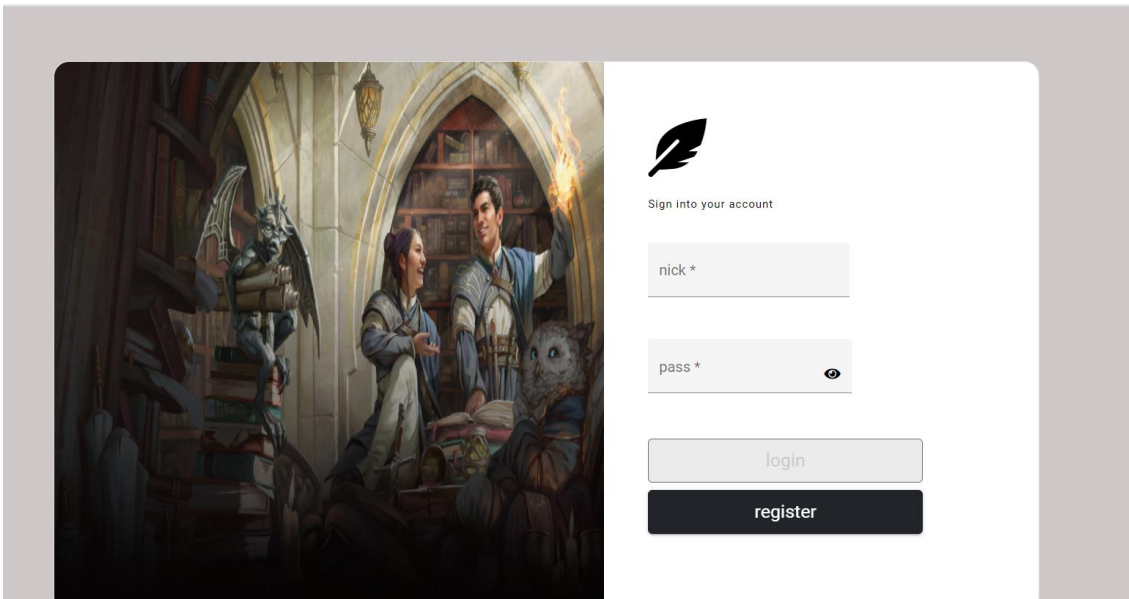
El despliegue inicial ha sido mediante H2 una base de datos en ejecución.

## Manual:

La web inicia directamente en la ventana de login, donde podemos decidir si logarnos con el usuario que tengamos o registrarnos.

El botón de login no se habilitará hasta que no se rellenen ambos campos, y pulsando sobre el ojo se revelará la contraseña

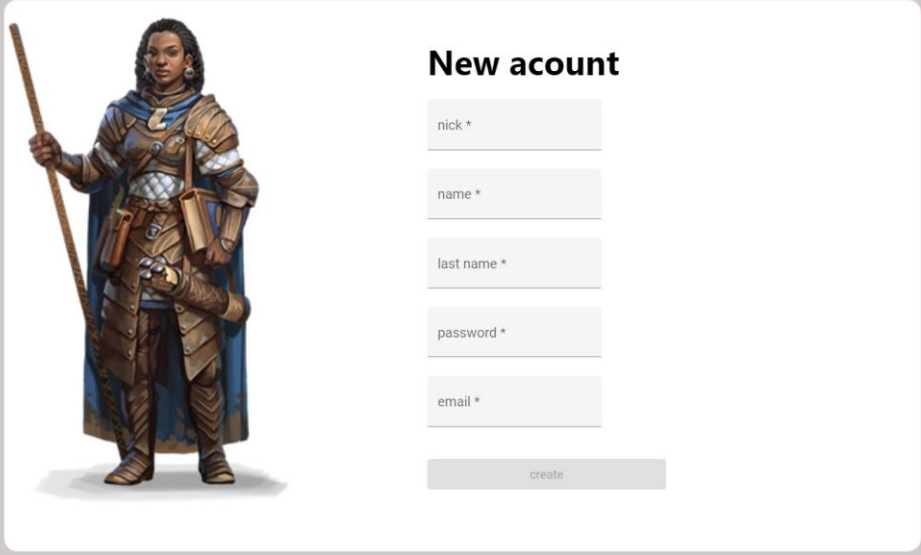
### LOGIN



The login page features a fantasy-themed illustration on the left and a login form on the right. The illustration depicts a man and a woman in a library, with a dragon on a staircase and a cat. The login form includes a feather logo, the text "Sign into your account", input fields for "nick \*" and "pass \*" (with a toggle eye icon), and "login" and "register" buttons.

Si decidimos registrarnos, se nos mostrara la siguiente pantalla donde se nos solicitara el Nick, nuestro nombre y apellido, una contraseña y el correo electrónico.

## REGISTER




The registration form is titled "New account" and is located on the right side of a white card. To the left of the form is a 3D illustration of a female paladin character in full plate armor with a blue cape, holding a long wooden staff. The form contains five input fields, each with an asterisk indicating it is required: "nick \*", "name \*", "last name \*", "password \*", and "email \*". Below these fields is a grey button labeled "create".

Hasta que todos los campos requeridos no se rellenen el botón de create no se activara, una vez insertado todos los datos al pulsar el botón si estos son correctos nos mandara de regreso a la ventana de login.

Una vez logados iremos a la ventana de vista de usuario, donde a medida que tengamos más personajes estos se nos mostraran, con su nivel nombre y clase.

## VISTA USUARIO

[Login](#) [New character](#) [my characters](#) perico 

### MASTER'S TRICK

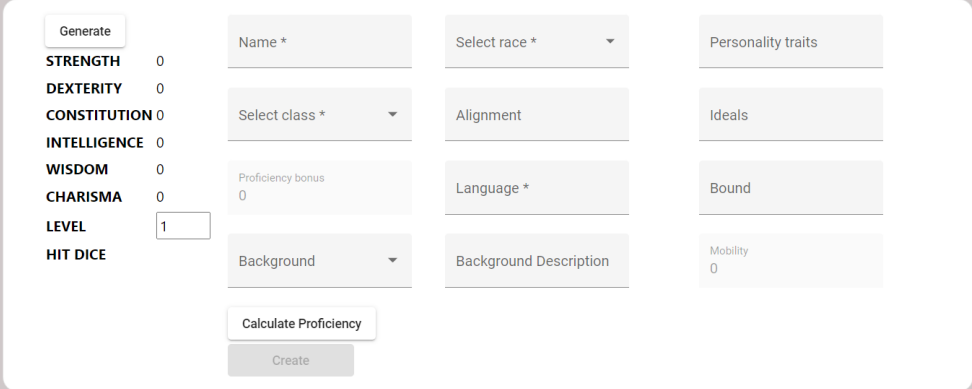
level	name	class
1	baigla	paladin
1	rosalia	cleriga

Si pinchamos sobre New character, iremos a la creación de personaje. Ahora bien, para calcular las características con pulsar en generate se crearán todas en base a 4d6 excluyendo el más bajo, level por defecto será 1 pero se puede editar, hit dice lo determina la clase que elijamos. Con el botón calculate proficiency nos calculara el bonus de

porvenir en base al nivel del personaje menos 1 entre 4(todo esto según el manual de creación de personajes de DND).

Cunado seleccionamos la raza nos pondrá automáticamente una lengua por defecto igual a la raza elegida, además se nos mostrará la movilidad también determinada por la raza.

## CREACION DE PERSONAJE



The screenshot shows a character creation form with the following fields and controls:

- Generate** button
- Attributes:** STRENGTH 0, DEXTERITY 0, CONSTITUTION 0, INTELLIGENCE 0, WISDOM 0, CHARISMA 0.
- LEVEL** input field with value 1.
- HIT DICE** label.
- Name \*** text input.
- Select race \*** dropdown menu.
- Select class \*** dropdown menu.
- Proficiency bonus** 0.
- Alignment** text input.
- Language \*** text input.
- Background** dropdown menu.
- Background Description** text input.
- Personality traits** text input.
- Ideals** text input.
- Bound** text input.
- Mobility** 0.
- Calculate Proficiency** button.
- Create** button.

Los valores de alignment, background con su descripción, traits, ideals y bound no son obligatorios. Esto lo remarco ya que solo cuando se rellenen los campos que son obligatorios se habilitará el botón de create y se nos desplazara automáticamente a la vista del perfil.

## Conclusiones:

El proyecto resultante ha sido una versión muy reducida de la idea inicial, el infravalorar la dificultad de organizar un proyecto sumado a comprender una framework entero han sido la causa principal de esta forzada reducción del proyecto.

El no poder añadir al menos dos juegos distintos ha sido una autentica decepción, el uso de angular es algo superficial para todo lo que puede llegar a ofrecer.

La interfaz resultante a sido mucho mejor de la que se esperaba.

De cara al futuro me gustaría reestructurar toda la parte de angular para poder sacarle muchísimo más partido. Luego reestructurar el backend de DND para que soporte correctamente posibles cambios de estructura en el juego, y si la mecánica o formatos del juego cambian poder actualizar con facilidad la aplicación.

Por ultimo y de cara a muy en el futuro antes de añadir otros juegos, preferiría añadir soporte para partidas de rol, ya que esto haría que la calidad y utilidad de la herramienta creciera exponencialmente.

## Índice de tablas e imágenes:

1. Maqueta Login.....	6p
2. Maqueta registro.....	7p
3. Maqueta vista usuario.....	7p
4. Maqueta creación p.....	8p
5. Maqueta vista p.....	8p
6. Diagrama app.....	9p
7. Manual login.....	12p
8. Manual Registre.....	13p
9. Manual vista usuario.....	13p
10. Manual creación personaje.	14p



## Bibliografía:

<https://angular.io/>

lista entra

<https://www.youtube.com/playlist?list=PLU8oAlHdN5BnNAe8zXnuBNzKID39DUwcO>

videos del 28 al 32

<https://www.youtube.com/playlist?list=PLy7TtEmBFuslZQ0f1MTYQXbsjeYUrSeMA>

<https://jasonwatmore.com/post/2019/06/10/angular-8-user-registration-and-login-example-tutorial>

<https://www.blog.duomly.com/angular-tutorial/#7-call-api-from-component>

Uso de tareas realizadas en bootcamps durante las practicas.